|  |
| --- |
| 软件工程基础及实践 |
| Python学习和开发环境的建立 | |
| 姓名：邴一伟  学号：03013413 |

|  |
| --- |
| biyiwei  [日期] |

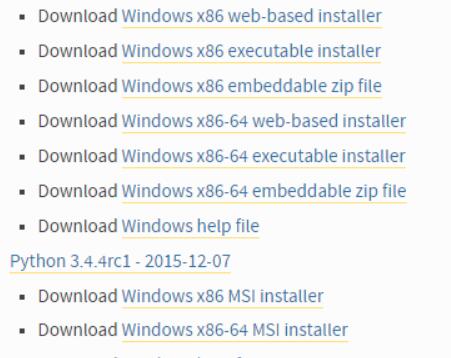
Python（英国发音：/ˈpaɪθən/ 美国发音：/ˈpaɪθɑːn/）, 是一种[面向对象](http://baike.baidu.com/view/125370.htm" \t "_blank)、解释型[计算机程序设计语言](http://baike.baidu.com/view/2561555.htm)，由[Guido van Rossum](http://baike.baidu.com/view/2975166.htm)于1989年发明，第一个公开发行版发行于1991年。

Python解释器、标准库及扩展包构成Python的基本开发环境，满足学习和软件开发的要求。Python解释器及其标准库来自班级分享；其他扩展包可用pip命令等方法安装。

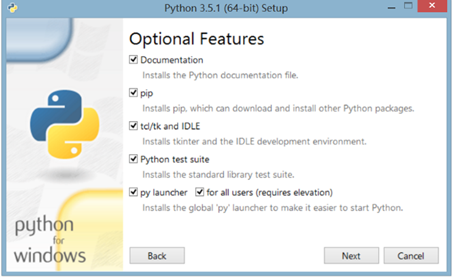
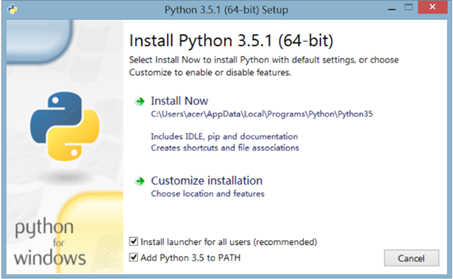
大型软件开发宜采用集成开发工具，常用的有: Eclipse，VS2015等，其中，Eclipse加PyDev插件的方式， 为较多Python程序开发人员使用。此外，还有使用编辑器(如Atom)构建开发环境的方法。

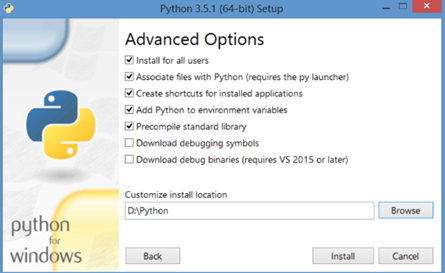
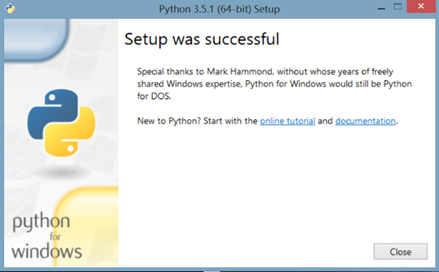
一、Python基本开发环境的建立

从官网<http://www.python.org>处下载

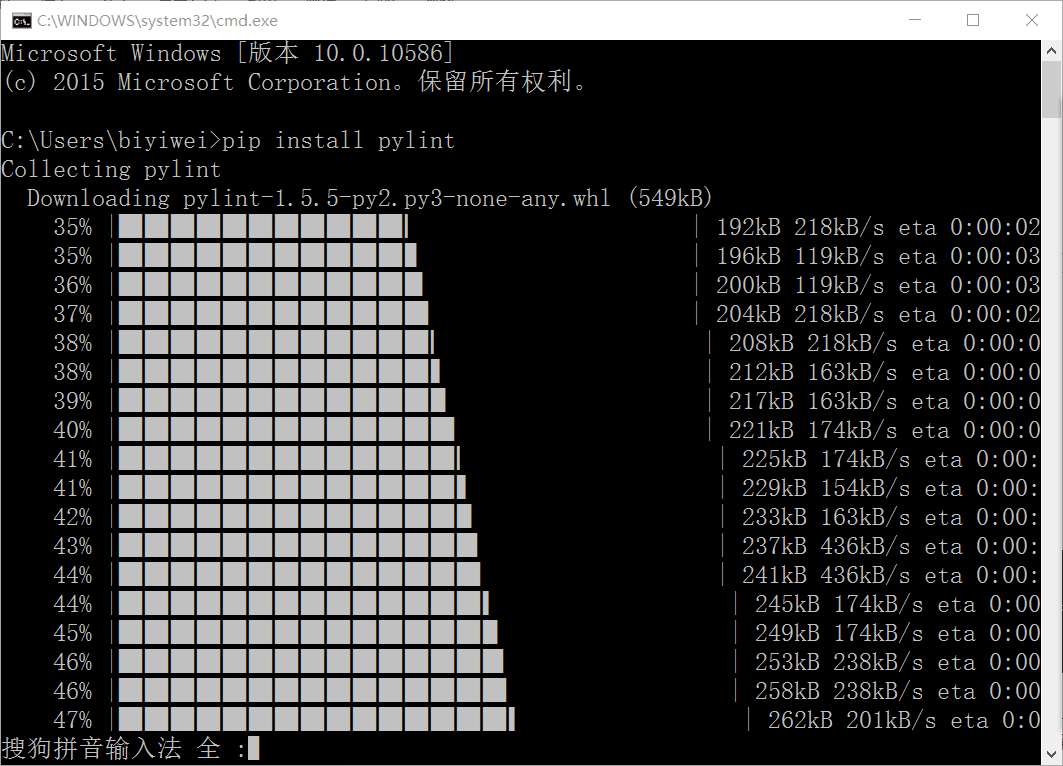
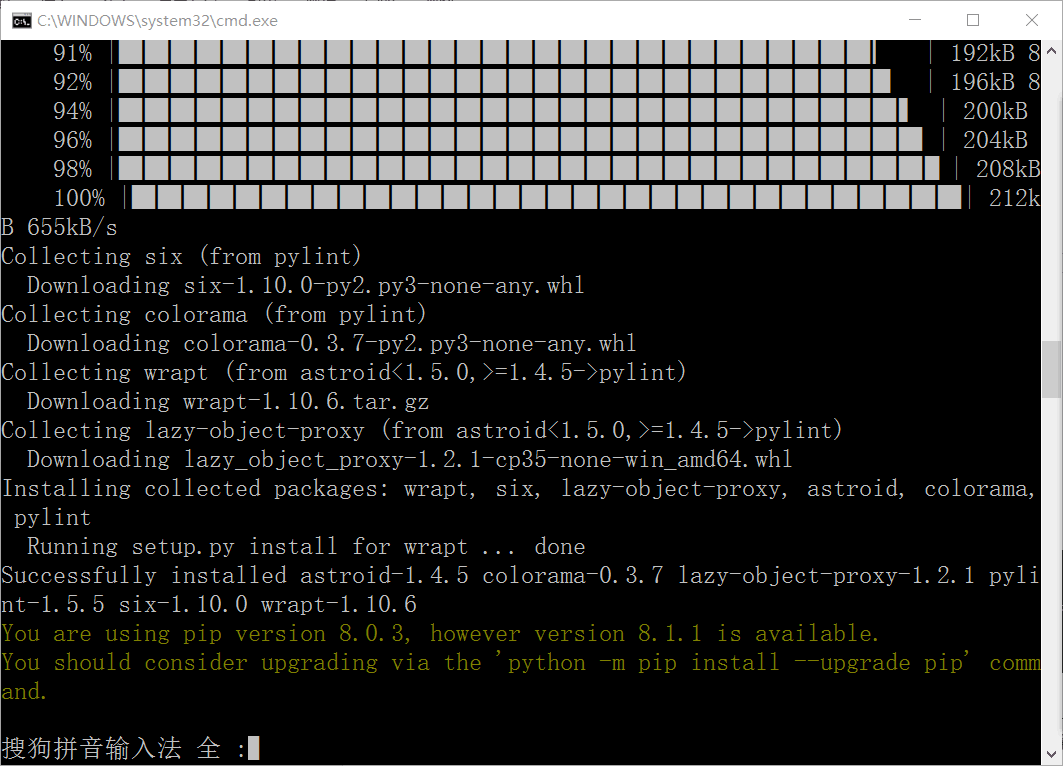


Windows系统安装： 1) 加环境变量PATH;

 2) 自定义安装目录： C:\User\biyiwei

然后，命令行(cmd)下更新pip到新版本：

python -m pip install -U pip

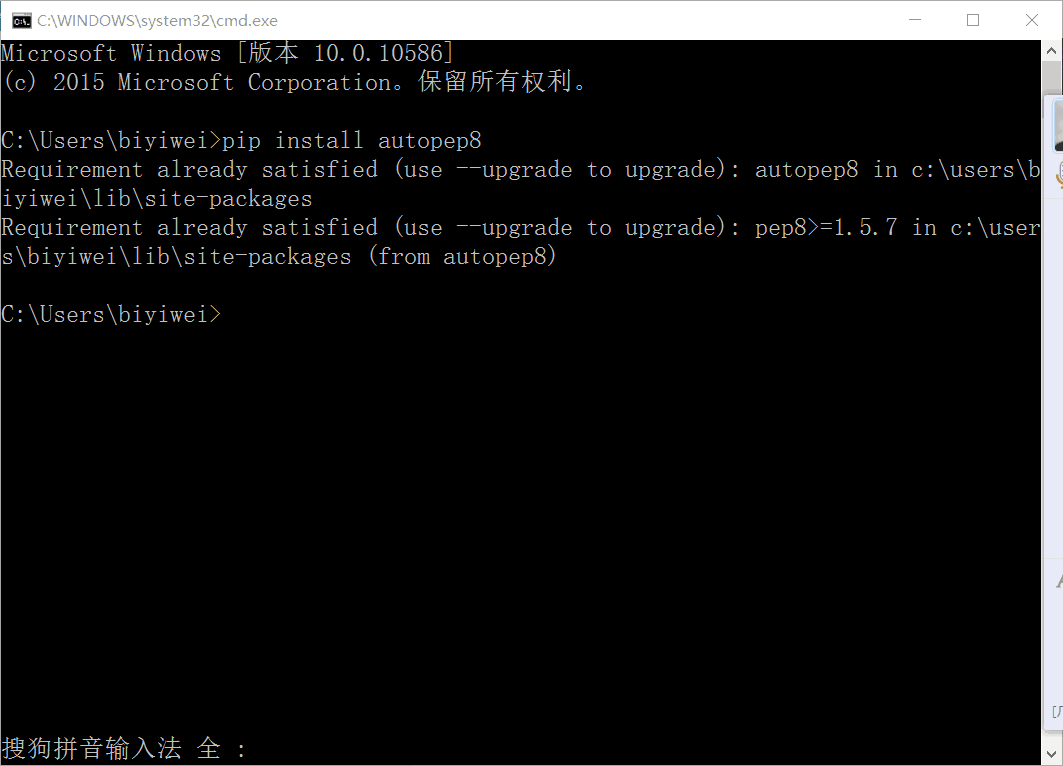


二、 Python扩展包安装

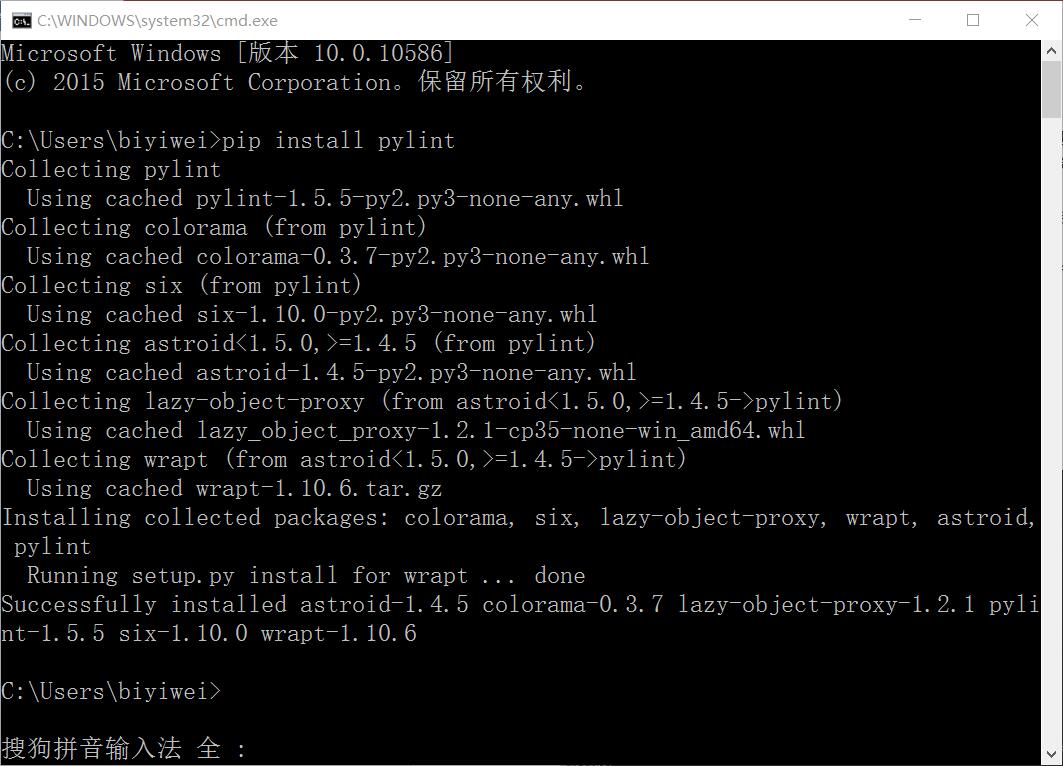
pip命令是软件包的基本安装方式，不过，有些软件包使用pip安装时，需要本地编译，编译条件如果不满足安装就会失败。

**2.1 编码规范**

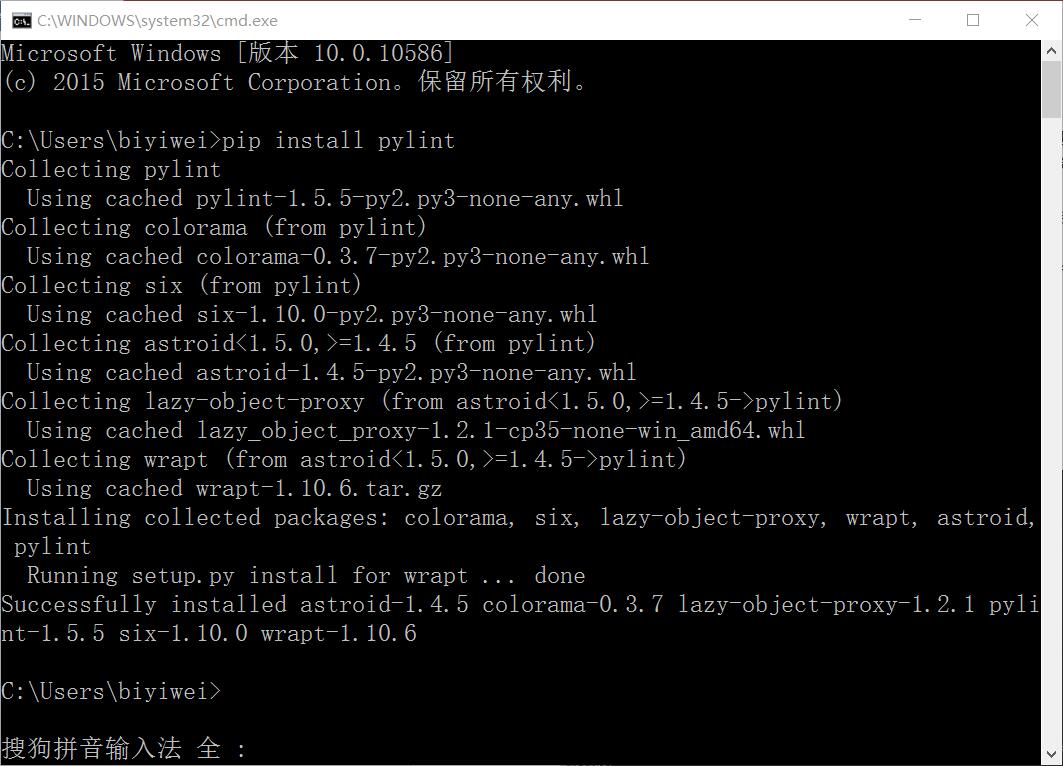
pip install autopep8



pip install pylint

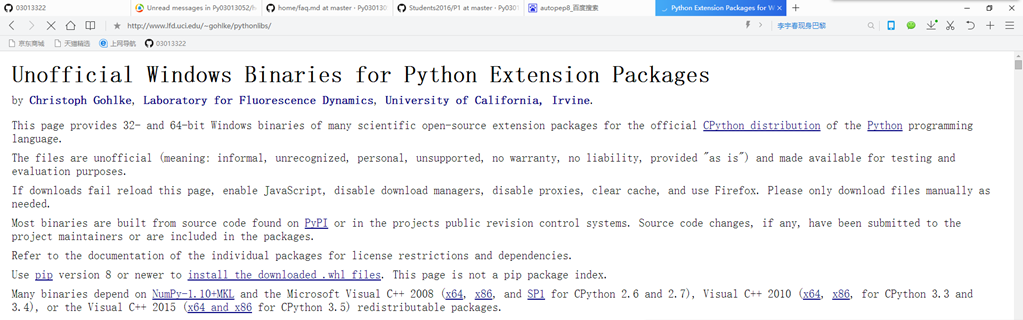


pip install pylint



**2 科学计算软件包**

**2.2.1 SciPy**

Scipy官网：<http://www.scipy.org/>,提供有全面的科学计算软件包信息。基本应用安装以下3个即可：

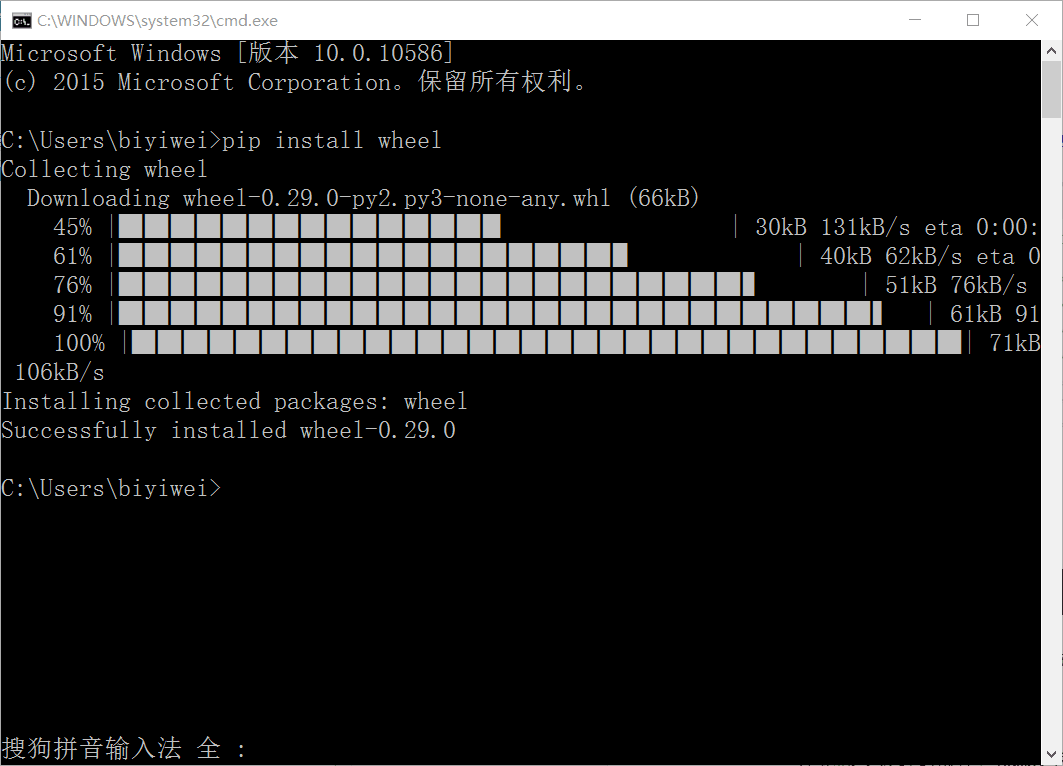
numpy

scipy

matplotlib

之后在命令行下输入

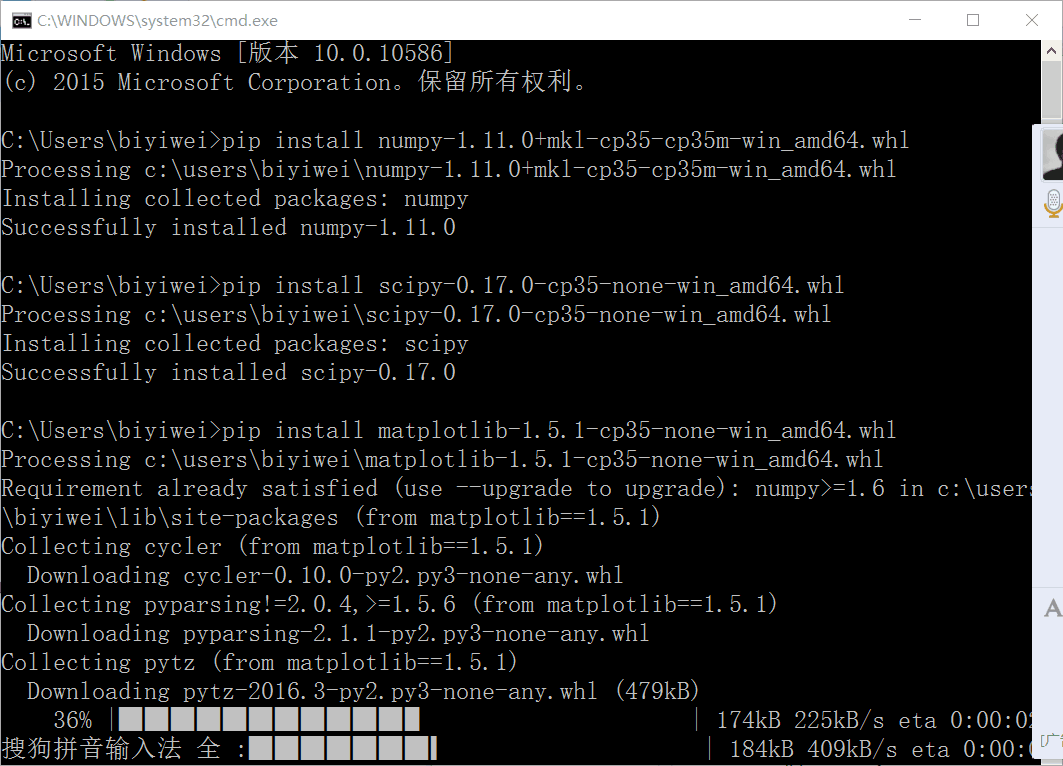
pip install wheel



Window环境下Scipy 软件包的 安装，建议从加州大学欧文分校下载编译好的\*.whl，然后，用pip逐个本地安装：

>pip install \*.whl

示例安装scipy：找到下载whl文件所在位置，然后shift+鼠标右键在此处打开命令窗口，键入>pip install scipy-0.16.1-cp35-none-win\_amd64.whl

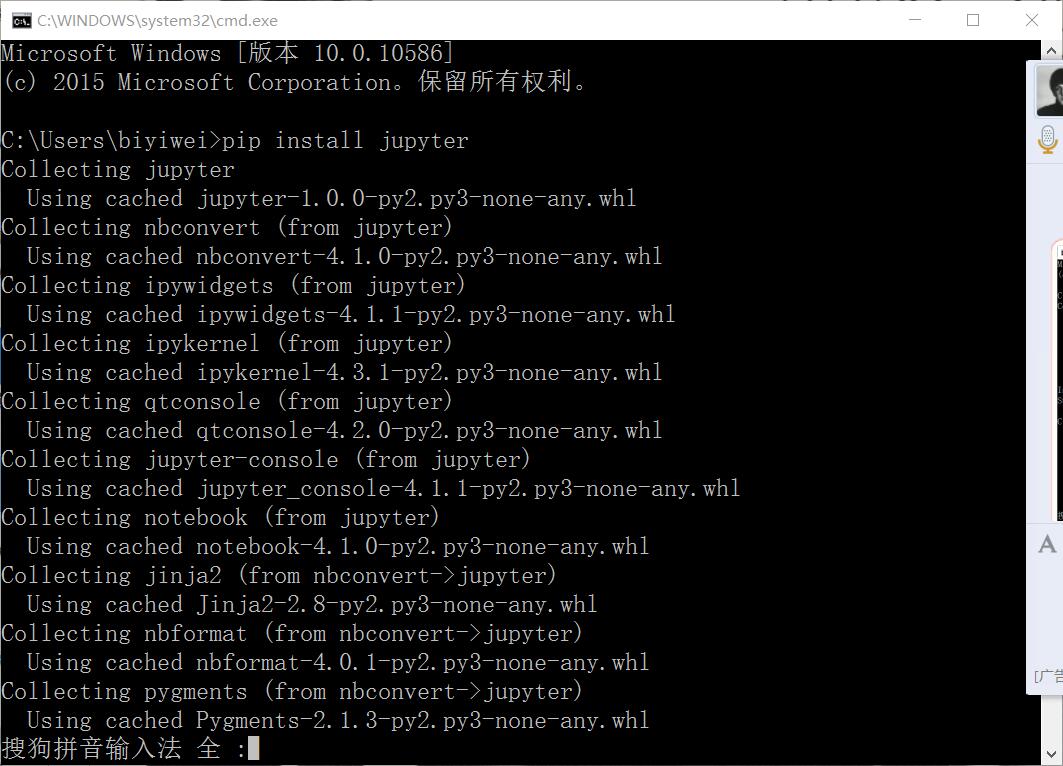


**2.2.2 交互计算 Jupiter**

1. **安装Jupiter**

**在命令行输入**

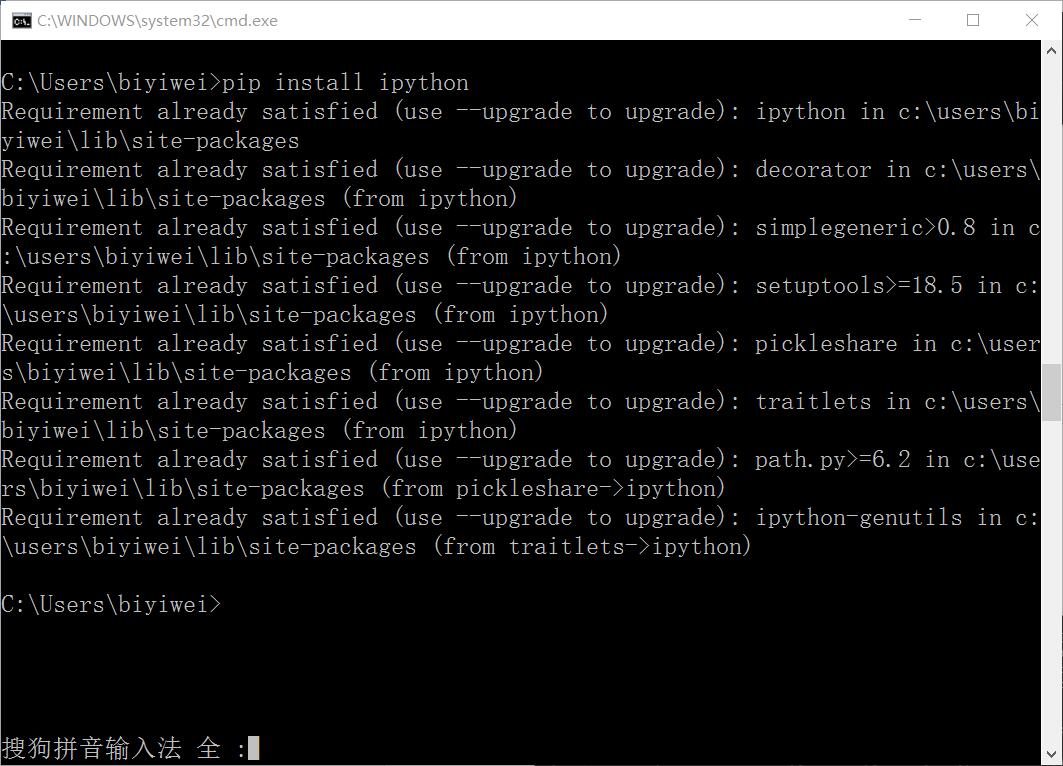
**pip install jupyter**



**2） 安装Python语言内核**

**在命令行输入**

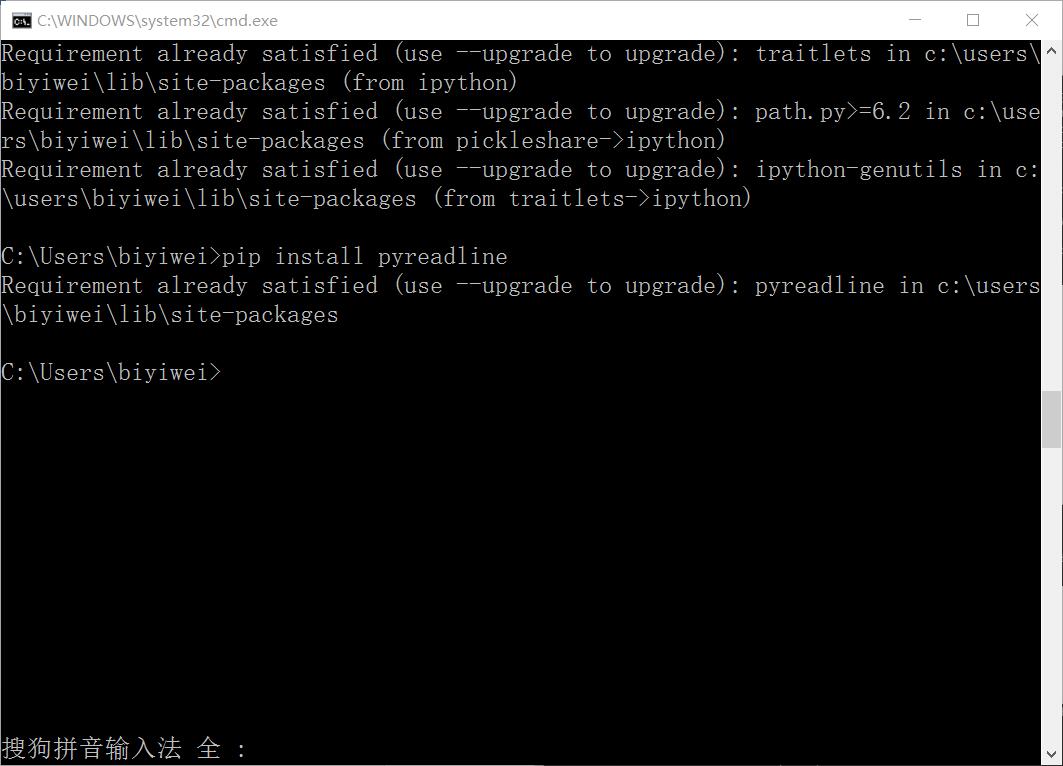
**pip install ipython**



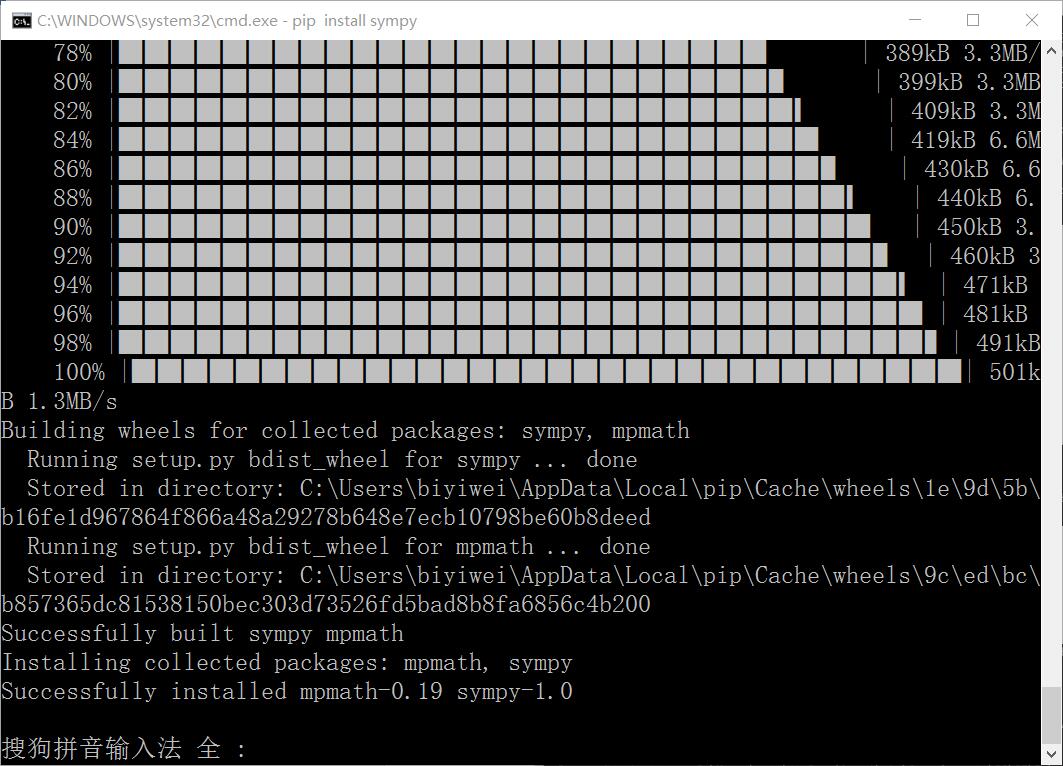
3)安装依赖包

在命令行输入

pip install pyreadline



pip install sympy



4）支持显示数学符号、公式，安装MathJax

在命令行输入

IPython

打开一个IPython的shell，然后，在其中键入如下代码

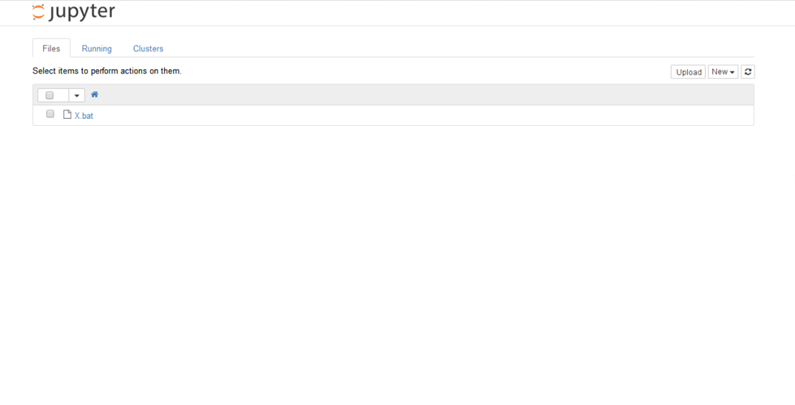
from IPython.external.mathjax import install\_mathjax

install\_mathjax()

5）运行notebook

在iPython notebook文件所在目录下，在命令行输入

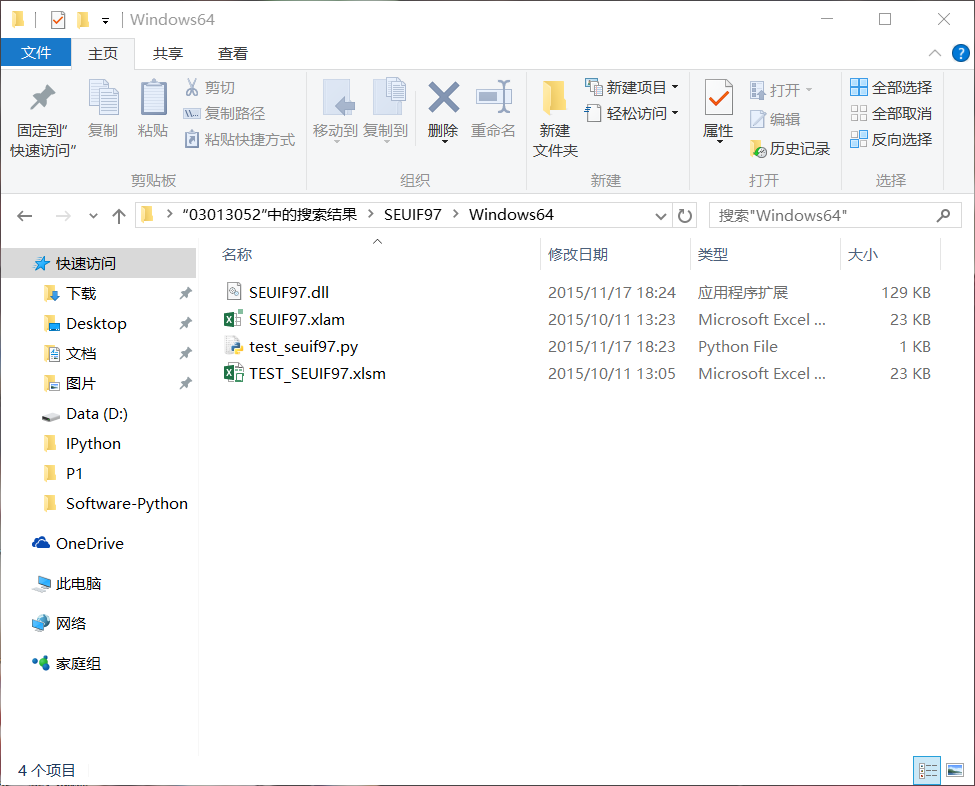
jupyter notebook

****

**2.2.3 IF97 for Python**

Windows 32/64位版: 从

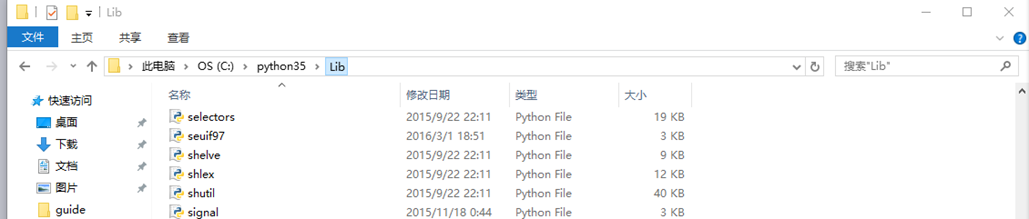
<https://github.com/Py03013052/SEUIF97>



下载：SEUIF97.dll和seuif97.py，或者从老师分享的文件中

1) SEUIF97.dll拷贝到 c:\windows\system



2) seuif97.py拷贝到 c:\python35\Lib（如果安装时安装在了其他文件夹就拷贝到安装文件夹的lib文件夹中）

1. 基于Eclipse的Python开发环境

3.1 安装Eclipse IDE

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台，专注于为高度集成的工具开发提供一个全功能的、具有商业品质的工业平台。Eclipse只是一个框架和一组服务，用于通过插件组件构建开发环境。Eclipse 附带了一个标准的插件集，包括 Java 开发工具（Java Development Tools，JDT）。Eclipse 开源，而且一般Java开发更倾向于Eclipse，使用安装免费，操作简单，且包含很多插件，使用方便，但程序启动较慢。

Eclipse IDE是使用Java开发的，电脑中需要预先安装好Java JRE/JDK软件包，因此，安装前，在命名行下输入

java –version

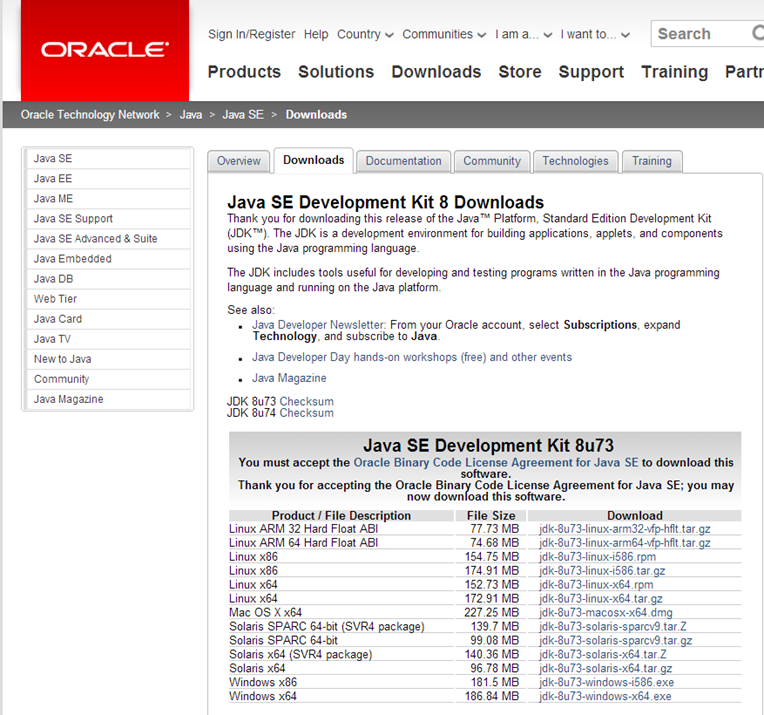
检查是否已经安装了Java软件包。

如果电脑中已经安装了Java ,会显示有关版本。

如果没有安装java 或者版本不能满足Eclipse IDE最低要求，需要从Oracle下载Java包。

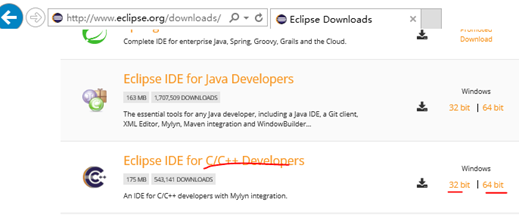
Java JDK的Oracle官方下载地址如下：

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>



Eclipse CDT官方下载地址：

<http://www.eclipse.org/downloads/>



根据操作系统32/64位，下载相应的版本，然后将下载的Eclipse CDT解压到指定目录下，运行解压目录下的：eclipse.exe即可。

如果使用Windows7 以上版本操作系统，建议将运行eclipse.exe，固定到任务栏。（在eclipse.exe文件名上，点鼠标右键即可）

3.2 安装Python开发插件PyDev

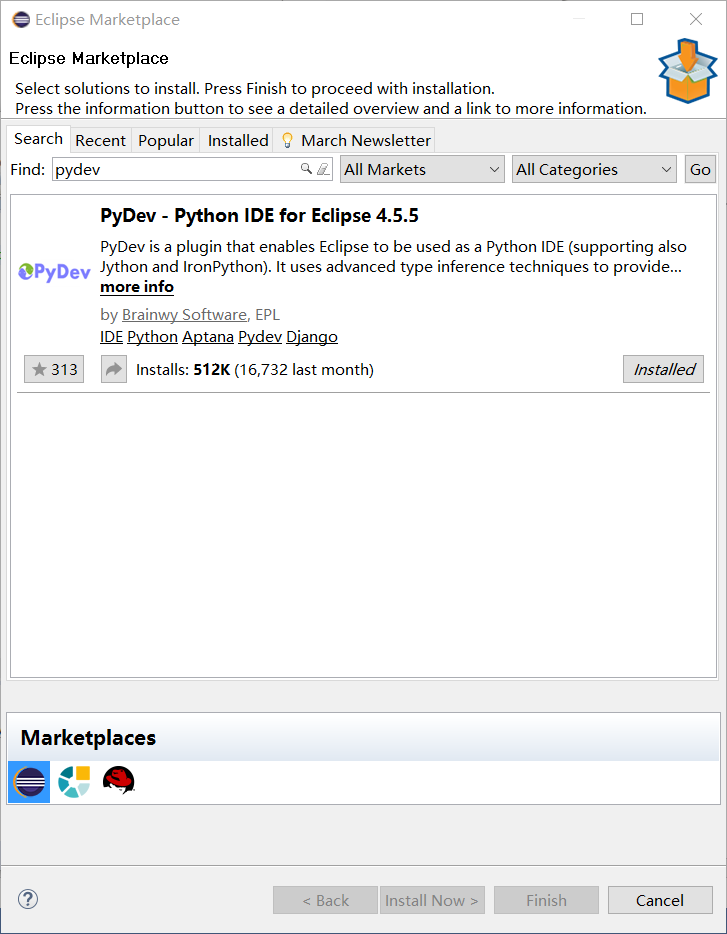
使用Eclipse IDE作为Python开发环境，需要：

1） 安装PyDev插件；

2）配置解释器为Python3.4；

*3.2.1 安装PyDev插件*

通过Windows->Eclipse Marketplaces进入市场，输入Pydev，找到Pydev安装/更新项目，在线安装即可：

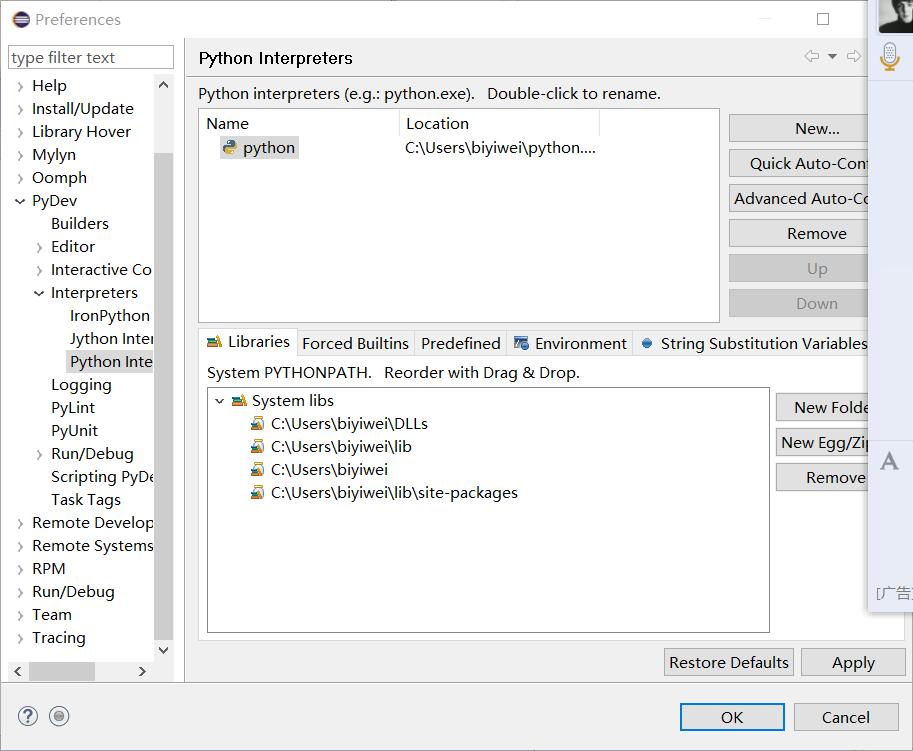


*3.2.2 配置 Python解释器*

安装好后，重新启动。通过：

Windows->Preference->Pydev->Interperters->Python Interperter

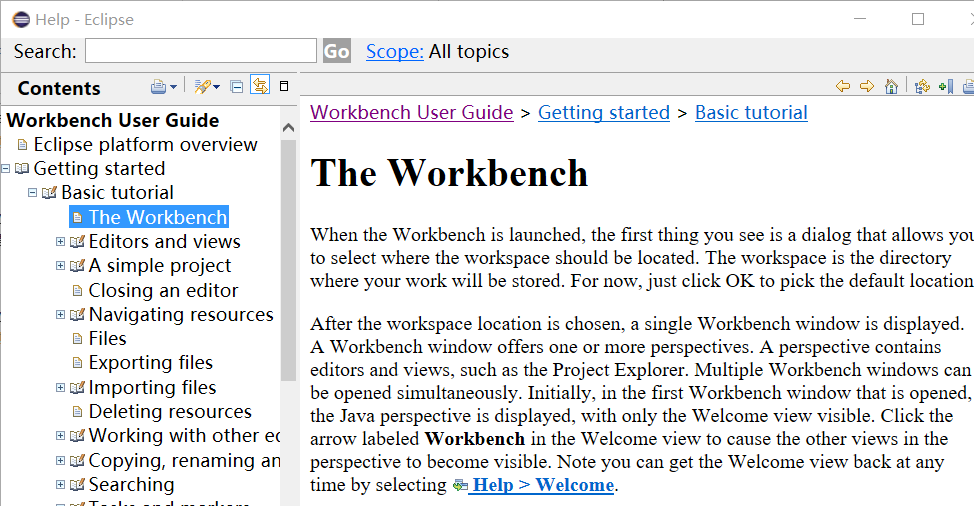
点其中的：Advanced Auto-config配置开发使用的 Python解释器版本



配置好后切换到Python场景，就可以进行Python开发了。

3.3 联机指南

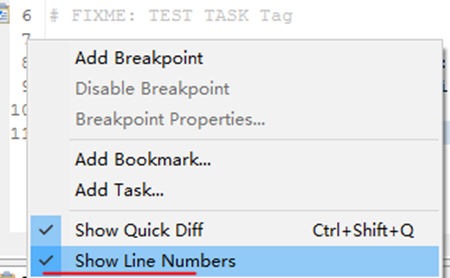
Help -> Help Contents



四、配置 PyDev提高效率

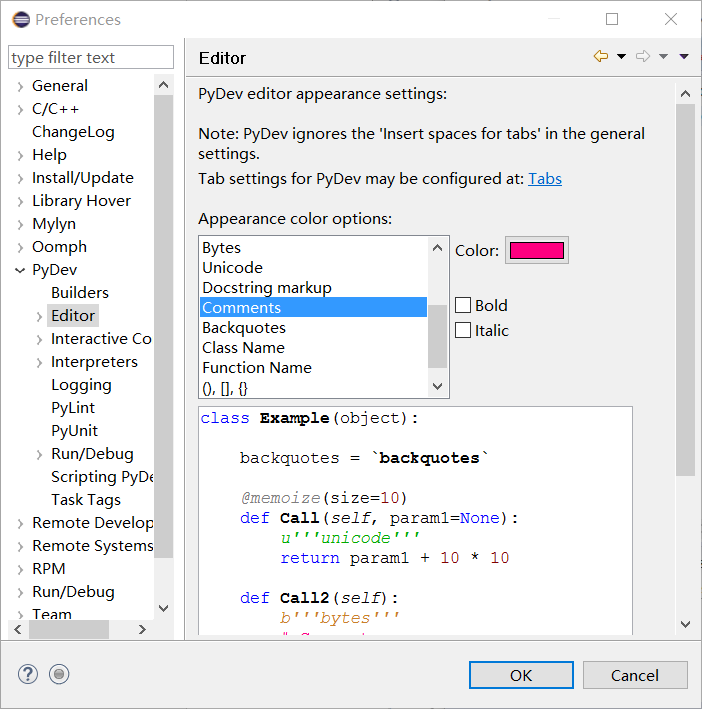
**4.1 显示源码行号**

右键源码的左边缘，选中“Show Line Number



**4.2 修改注释颜色提高可读行**

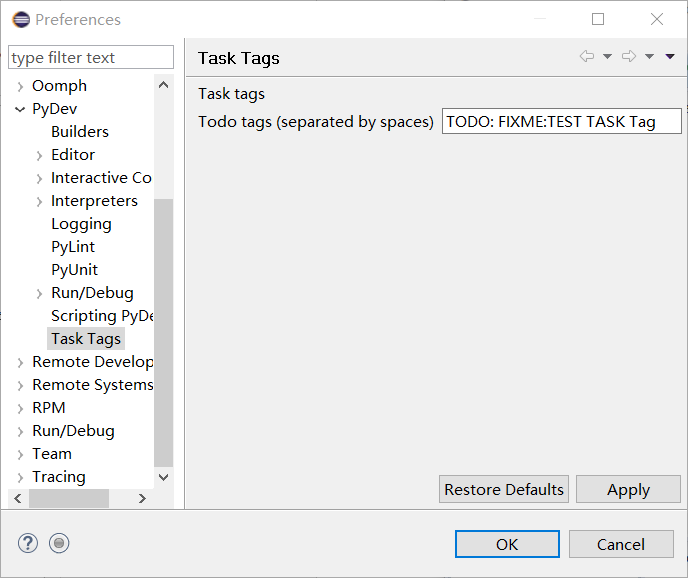
从Window → Preferences→PyDev→Editor,进入配置界面：



**4.3 配置任务标签**

在程序开发过程中可以在代码中标识，当前任务状态，计划开发工作。在代码中标识任务，可以使用任务标签，然后，让开发环境识别到工作空间的任务列表中。

1. 配置任务标签：



可以根据不同的任务类型，自己增加自定义标签

2）添加任务标签注释

在程序源码中加入使用任务标签的注释：

3）识别任务

保存新修改、运行程序一次或者选择 Project → Clean，

就可将使用任务标签注释加入任务列表。如果当前任务窗口可视，刚加的任务，就会立刻显示在任务窗口中。

五、基于PyDev的语言规范静态检查

PyDev中集成了PEP8,AutoPEP8 和Pylint代码检查功能，这些功能默认状态都是关闭的。

程序开发过程中，要有规范意识，但不可能有很高的规范性，过分注意规范会影响开发进程。这时如果一直开启代码规范检查，经常提示不规范，会对开发形成负面影响，所以，默认关闭是合适的。在程序开发一个阶段结果出来时，进行规范性检查更好。

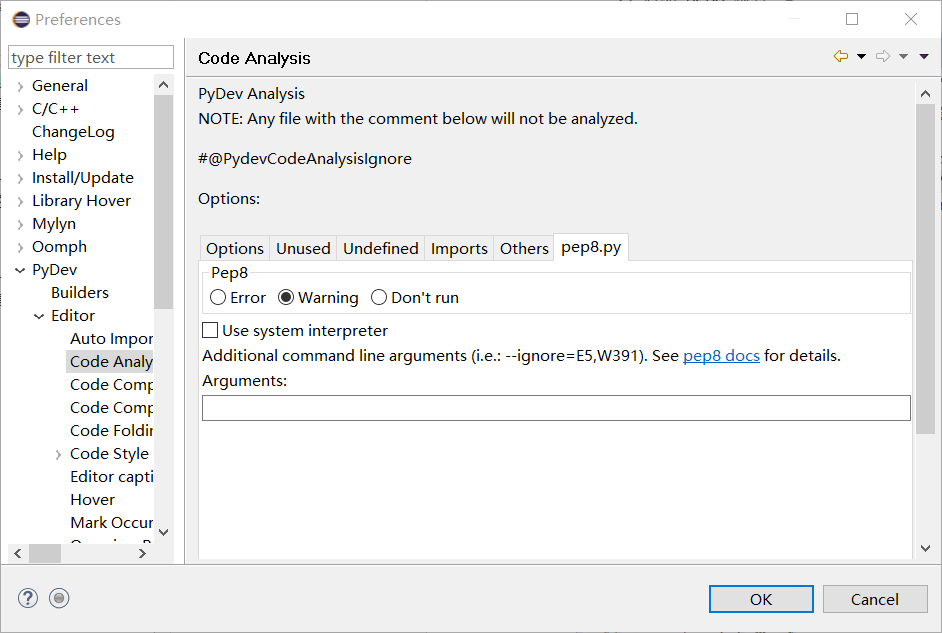
5.1 PEP8检查和修改

1）启动 pep8 检查：

Window > Preferences

PyDev > Editor > Code Analysis > pep8.py

选择Errors/Warnings其中之一。

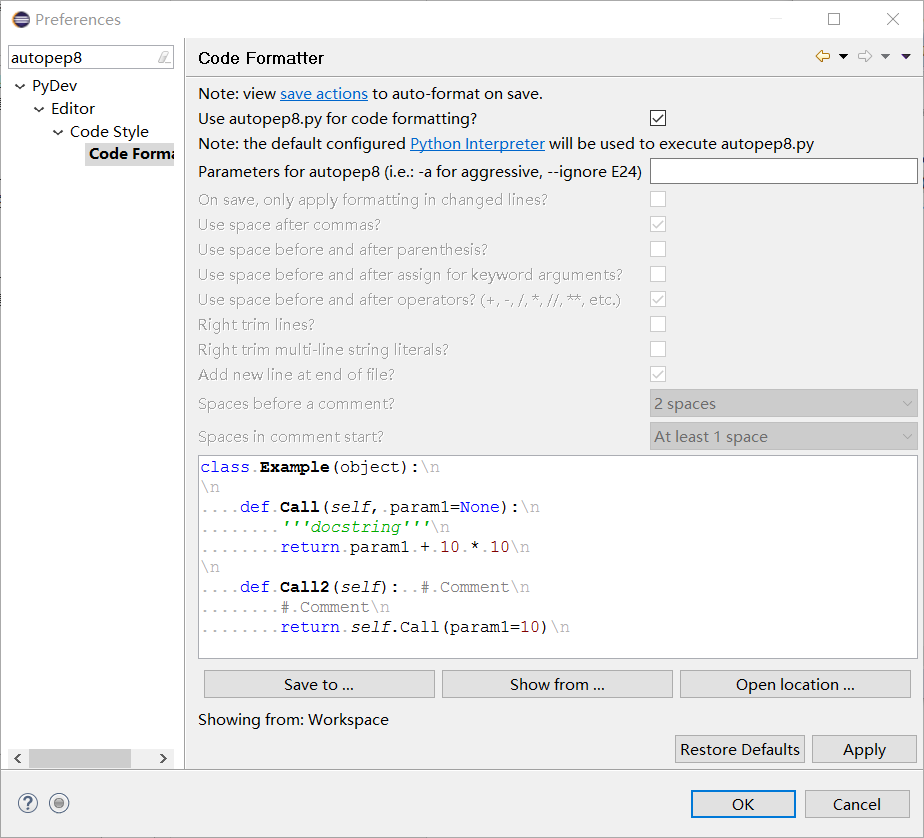


右键Python工程，选择 PyDev，点 "code analysis"，即可对工程中所有Python源码进行PEP8检查： .

2）启动autopep8自动修改：

点Windows -> Preferences ->输入 'autopep8' 作为搜索串.

选择（Check）: Use autopep8.py for code formatting?



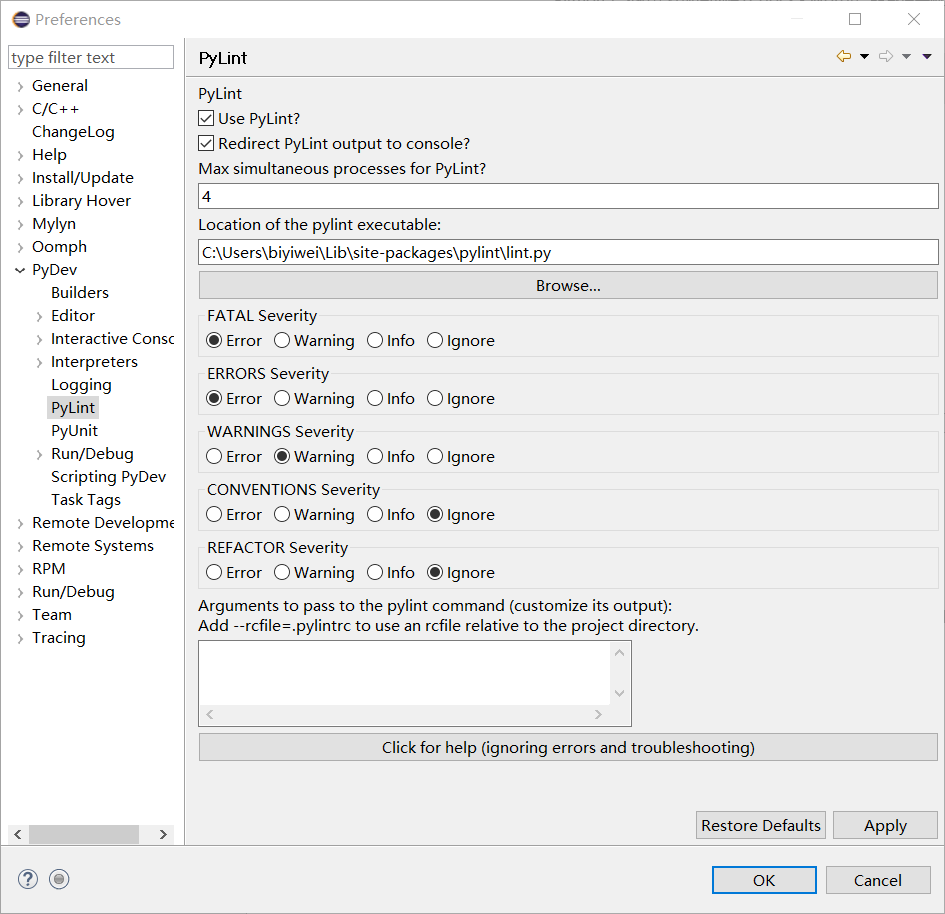
在Python源码窗口，按 CTRL-SHIFT-F 就可以自动修改代码

5.2 使用Pylint

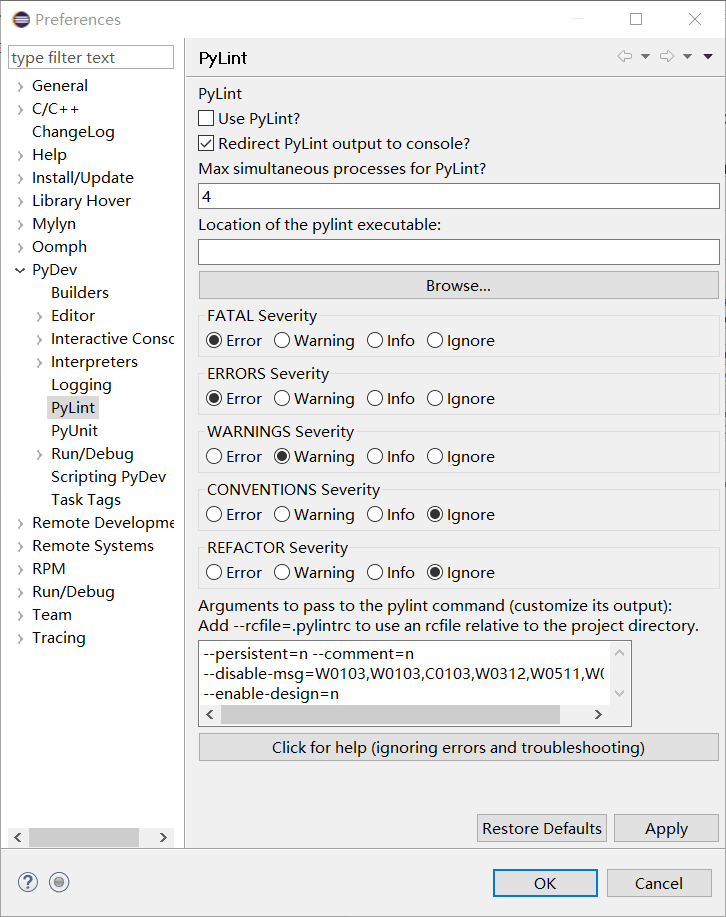
PyDev 默认不开启Pylint。通过

Window -> preferences -> Pydev -> Pylint,选中"Use pylint?",

找到安装好的lint.py的地址,例如”C:\Users\biyiwei\Lib\site-packages\pylint\lint.py”



配置参数，限制Pylint的输出



选中Project->Build Automatically,这样程序修改，保存时pylint就会自动检查项目中的代码,也可用Ctrl+B手动build触发pylint。